REMARKS

In an Office Action dated October 3, 2007, the Examiner re-opened prosecution following applicants' appeal herein; rejected claims 1, 2, 5, and 7 under 35 U.S.C. §103(a) as unpatentable over Lindsey (US 5,896,536); and rejected claims 4, 8-12 and 25-32 under 35 U.S.C. §103(a) as unpatentable over *Lindsey* in view of Wygodny (US 6,282,701 B1).

Applicants have amended certain independent claims herein to further clarify the scope of the claimed invention, if such clarification be necessary. In particular, independent claims 1, 25 and 28 are amended to recite that the trigger expression results in an L value during execution of the program or process (this limitation was already in independent claim 12). Dependent claim 5 has been cancelled as superfluous. Independent claims 1 and 25 are further amended to recite that monitoring is performed without alteration of the logic flow of the monitored program. While applicant believes this latter limitation is unnecessary, the necessary limitations being already inherent in the claims, it is added for further clarification, if such clarification be necessary. As amended, the claims are patentable over the cited art.

Applicants have discussed the operative features of their invention and the subject references at length in the appeal brief and previous responses filed herein, and applicants incorporate by reference those discussions for a background understanding of the issues involved herein, without necessarily repeating verbatim everything said in the previous papers.

In response to the previous rejections, applicants pointed out a significant difference between *Wygodny* and their claimed invention being the nature of the triggering event. I.e., in accordance with applicants' invention, a triggering event is a reference to a data value as opposed to execution of an instruction. Although an instruction may cause a reference to a data value, in

Wygodny it is the instruction that is the trigger. The programmer specifies an instruction for tracing, and encountering that instruction causes trace data to be collected.

In response to the previous rejection, applicants allowed themselves to discuss the high-level abstractions of the claims at issue, because there were fundamental differences between the way in which applicants' claimed invention worked and that in which *Wygodny* worked. At that time, the discussion between applicants and the Examiner appeared to have bogged down into a discussion of what occurs when a machine-level instruction executes. E.g., instructions reference memory locations, which can represent state data, which can trigger traces, etc. Applicants hoped that the higher-level abstract explanation might shine some light on this subject.

Apparently, the Examiner has taken applicants' remarks in response to the previous rejection to heart, and shown applicant a reference in which analogous underlying abstract concepts are disclosed. This is interesting, but it is a long way from showing the claim limitations.

In the present office action, the Examiner relies primarily on *Lindsey*. *Lindsey* was previously cited, but the rejection was applied differently, and therefore applicants respond more particularly to the current rejection.

Lindsey discloses a technique for debugging object-oriented programs, in which the activity of particular object instances is recorded. *Lindsey* refers to this activity as "tracing", although it is somewhat different from tracing in the sense of tracing machine-level instructions. As disclosed by *Lindsey*, in certain circumstances, it may be desirable to understand the activity of particular object instances of multiple objects of the same class in an object-oriented program. If the methods applicable to the class at large are traced (using traditional techniques), then the trace may not be sufficiently specific. In order to understand the behavior of particular object

instances, *Lindsey* proposes to modify the program logic so that one or more special methods are created as substitutes for the base class methods, the special methods being used only for the particular object instance which it is desired to "trace". These special methods can then generate messages to a "trace manager", which trigger collection of state data.

It is true that, at a high level of abstraction, *Lindsey* discloses collecting state data based on the activity of data as opposed to program instructions. But the claims recite something more than this high level concept.

Applicants' claim 1, as amended, recites:

1. A method of tracing the activity of an expression, said method comprising the machine-implemented steps of:

(a) receiving, from a user, a specification of a machine-implemented process in which a *trigger expression resulting in an L value during the machine-implemented process* is to be traced;

(b) receiving, from a user, a specification of the trigger expression to be traced in the machine-implemented process, said *trigger expression representing a non-executable data value having a state*;

(c) responsive to steps (a) and (b), *monitoring execution of said machine-implemented process* ***without altering a logic flow of said machine-implemented process*** *to detect occurrences of a plurality of references to a location in machine memory representing a state of said trigger expression*, wherein each said occurrence of a reference to a location in machine memory representing a state of said trigger expression occurs as a result of executing said machine-implemented process;

(d) responsive to each detected occurrence of a reference to said location in machine memory representing a state of said trigger expression, storing the respective state of the trigger expression at the time of the respective detected occurrence of a reference to said location in machine memory representing a state of said trigger expression to create a history of said trigger expression within the machine-implemented process, said storing step being performed without interrupting the machine-implemented process; and

(e) restoring the state of the trigger expression when requested. [emphasis added]

Independent claim 25, while not identical in scope, contains analogous limitations to those italicized above. Independent claims 12 and 28 also contain analogous limitations, but without the language in bold text.

Lindsey does not disclose a monitoring process at all. Rather, *Lindsey* discloses a process whereby ***program logic flow is altered*** so that the activity of particular object instances in an object-oriented program cause messages to be generated, these messages then causing state data to be recorded by which the behavior of the object instance can be better understood.

Applicants disclose and claim a process of monitoring ***particular memory locations*** during execution of a computer program. These memory locations correspond to non-executable data values (L values) of a trigger expression which is specified as input by a programmer.

Applicants' "trigger expression", standing alone, is broad enough to encompass an object variable instance, and at some level that object variable instance is stored in a memory location. But *Lindsey* does not monitor access to that memory location, as recited in applicants claims. *Lindsey* alters the program logic to generate a record whenever certain methods are used to access the object variable instance.

There is nothing in *Lindsey* which teaches, suggests, or otherwise renders obvious the monitoring of particular memory locations corresponding to state variables, as claimed by applicants. Further, to the extent that *Lindsey* specifically discloses alteration of the program logic flow to create a record of activity for the object variable instance, *Lindsey* teaches away from the claim limitations.

Finally, it is unclear what, if anything, is taught or suggested by the hypothetical combination of *Wygodny* and *Lindsey*. Neither references discloses monitoring a particular

memory location corresponding to a trigger expression provided by the programmer, as recited in applicants' claims. Even if one assumes that the references are properly combinable, there is nothing that suggests any particular combination of elements or resultant device. *Lindsey* discloses specifying an object instance and replacing the methods which access that object instance with others. *Wygodny* discloses various options for collecting trace data triggered by particular code statements. The hypothetical combination, if it suggests anything at all, appears to suggest using *Wygodny*'s trace options to collect and analyze data provided by a replacement method as disclosed in *Lindsey*. Such a combination, even if suggested, fails to meet the limitations of applicants' claims.

For all of the reasons explained above, the claims as amended not obvious over Lindsey or *Lindsey* and *Wygodny* in combination.

In view of the foregoing, applicants submit that the claims are now in condition for allowance, and respectfully request reconsideration and allowance of all claims. In addition, the Examiner is encouraged to contact applicants' attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,

CARY L BATES, et al.

By:_____
Roy W. Truelson
Registration No. 34,265

Telephone: (507) 289-6256 (Office/Fax)
(507) 202-8725 (Cell)